A finite difference scheme for plate synthesis

A FINITE DIFFERENCE PLATE MODEL

Stefan Bilbao
Sonic Arts Research Centre
Queen's University Belfast

Maarten van Walstijn
Sonic Arts Research Centre
Queen's University Belfast

ABSTRACT

In this short paper, a simple flexible difference scheme for the simulation of rectangular plate vibration is presented, along with various implementation details (scheme coefficients, a discussion of boundary condition implementation, explicit stability bounds, and memory and computational requirements), and a discussion of features such as moving excitation and readout locations. Sound examples are presented.

1. INTRODUCTION

In this short article, we present a condensed but complete treatment of a simple finite difference scheme for plate vibration, including a discussion of many practical features of interest, such as frequency-dependent damping, moving input and output locations, spatialization issues, boundary termination and initialization, and computational complexity. Such models, though they are computationally intensive, offer a wide range of rich sounds to a composer, and the opportunity to create multichannel sounds with internal conherence (i.e., the various channels all derive from a single physical model). Even the computational requirements for full scale simulations such as these are rapidly becoming feasible even on desktop machines, and are comparable to requirements of other techniques (such as modal synthesis [1]). Time-domain simulation methods for plates in a musical sound synthesis context have been presented before, primarily by Schedin et al. [2] and Lambourg et al. [3]; the presentation here is aimed at a discussion of implemetation details, and issues of practical concern, particularly with regard to sound spatialization; in addition, we present an explicit simplified stability condition for the finite difference scheme.

2. A THIN PLATE MODEL

A plate model, suitable for musical sound synthesis applications, is a variant of the classical Kirchhoff model [4]:

$$\frac{\partial^2 u}{\partial t^2} = -\kappa^2 \nabla^4 u + c^2 \nabla^2 u - 2\sigma \frac{\partial u}{\partial t} + b_1 \frac{\partial}{\partial t} \nabla^2 u + f(x,y,t) \tag{1}$$

Here $u(x,y,t)$ is the transverse plate deflection, defined over the rectangular region $x \in [0, L_x]$, $y \in [0, L_y]$ and for time $t \geq 0$. ∇^2 is the Laplacian, and ∇^4 the biharmonic operator. Here, the stiffness parameter κ^2 is defined by

$$\kappa^2 = \frac{Eh^2}{\rho(1-\nu^2)}$$

where E, h, ρ and ν are Young's modulus, plate thickness, density and Poisson's ratio, respectively for the plate (all assumed constant here). The term involving the parameter c represents a contribution to the dynamics due to applied tension; indeed, if c is large relative to κ, the equation describes the behaviour of a membrane. There are two terms which give rise to loss: the term of coefficient σ controls the gross decay rate of plate oscillation, and that with coefficient $2b_1$ allows for higher rates of loss at high frequencies. The term $f(x,t)$ represents a driving term, time-varying, and possibly spatially distributed. The model above can be considered to be a simple generalization of that for a stiff string to two spatial dimensions.

The second order time-dependent PDE (1) requires two initial conditions, i.e., an initial displacement $u(x,y,0)$ and velocity $\frac{\partial u}{\partial t}(x,y,0)$. It also requires the specification of two conditions at any boundary; the only two conditions we will examine in this short paper are the so-called clamped and pinned conditions, given by

$$u = \frac{\partial u}{\partial x_n} = 0 \tag{2a}$$

$$u = \frac{\partial^2 u}{\partial x_n^2} = 0 \tag{2b}$$

where $\frac{\partial}{\partial x_n}$ represents a partial derivative in a direction normal to a given boundary. Free boundary conditions, which are more difficult to correctly pose in the case of thin plate theory, will not be discussed here.

3. FINITE DIFFERENCE SCHEME

In order to solve (1) numerically, we can make use of a finite difference approximation, first defining a grid function $u_{i,j}^n$ representing an approximation to the solution $u(x,y,t)$ at coordinates $x = \Delta_x i$, $y = \Delta_y j$, and $t = nT$. Here, Δ_x and Δ_y are the grid spacings in the x and y directions, respectively, and T is the time step ($1/T$ is the sample rate). Notice, in particular, that Δ_x and Δ_y are, in general, different; we define the parameter α by $\alpha = \Delta_x / \Delta_y$.

3.1. Difference Operators

Standard approximations to the first and second time differential operators are given by

$$\begin{aligned}
\delta_t^2 u_{i,j}^n &= \frac{1}{\Delta_t^2}\left(u_{i,j}^{n+1} - 2u_{i,j}^n + u_{i,j}^{n-1}\right) \approx \frac{\partial^2 u}{\partial t^2} \\
\delta_{t0} u_{i,j}^n &= \frac{1}{2\Delta_t}\left(u_{i,j}^{n+1} - u_{i,j}^{n-1}\right) \approx \frac{\partial u}{\partial t} \\
\delta_{t-} u_{i,j}^n &= \frac{1}{\Delta_t}\left(u_{i,j}^{n} - u_{i,j}^{n-1}\right) \approx \frac{\partial u}{\partial t}
\end{aligned}$$

The first and second operators above are centered and second-order accurate; the third, a backward difference operator, is first-order accurate. Second-order centered difference approximations to second derivatives in the x and y directions are given by

$$\begin{aligned}
\delta_{x0}^2 u_{i,j}^n &= \frac{1}{\Delta_x^2}\left(u_{i+1,j}^{n} - 2u_{i,j}^n + u_{i-1,j}^{n}\right) \approx \frac{\partial^2 u}{\partial x^2} \\
\delta_{y0}^2 u_{i,j}^n &= \frac{1}{\Delta_y^2}\left(u_{i,j+1}^{n} - 2u_{i,j}^n + u_{i,j-1}^{n}\right) \approx \frac{\partial^2 u}{\partial y^2}
\end{aligned}$$

and thus the Laplacian and biharmonic operators can be approximated by

$$\begin{aligned}
\delta_+^2 = \delta_{x0}^2 + \delta_{y0}^2 &\approx \nabla^2 \\
\delta_+^2 \delta_+^2 &\approx \nabla^4
\end{aligned}$$

3.2. An Explicit Finite Difference Scheme

At this point, we may substitute the above operators for their continuous time/space counterparts in (1), to get the following explicit finite difference scheme:

$$\delta_t^2 u = -\kappa^2 \delta_+^2 \delta_+^2 u + c^2 \delta_+^2 u - 2\sigma \delta_{t0} u + b_1 \delta_{t-} \delta_+^2 u + f \quad (3)$$

This can be rewritten as an explicit recursion in the grid function $u_{i,j}^n$, where an interior point in the domain (we will define this presently) is updated according to

$$\begin{aligned}
u_{i,j}^{n+1} &= \eta \sum_{|k|+|l|\le 2} \beta_{|k|,|l|} u_{i+k,j+l}^n \\
&\quad + \eta \sum_{|k|+|l|\le 1} \gamma_{|k|,|l|} u_{i+k,j+l}^{n-1} + T^2 \hat{f}_{i,j}^n
\end{aligned} \quad (4)$$

where $\hat{f}_{i,j}^n$ is an approximation to $f(i\Delta_x, j\Delta_y, nT)$ (to be discussed in section 3.2.4 below) and

$$\begin{aligned}
\beta_{0,0} &= 2 - 2\mu^2(3 + 4\alpha^2 + 3\alpha^4) \\
&\quad -2(\lambda^2 + \nu)(1 + \alpha^2) \\
\beta_{1,0} &= 4\mu^2(1 + \alpha^2) + \lambda^2 + \nu \\
\beta_{0,1} &= \alpha^2(4\mu^2(1 + \alpha^2) + \lambda^2 + \nu) \\
\beta_{1,1} &= -2\mu^2\alpha^2 \\
\beta_{2,0} &= -\mu^2 \\
\beta_{0,2} &= -\alpha^4\mu^2 \\
\gamma_{0,0} &= -1 + 2\nu(1 + \alpha^2) + \sigma T \\
\gamma_{1,0} &= -\nu \\
\gamma_{0,1} &= -\alpha^2\nu
\end{aligned}$$

and where we have also defined

$$\begin{aligned}
\mu &= \frac{\kappa T}{\Delta_x^2} \qquad \lambda = \frac{cT}{\Delta_x} \\
\eta &= \frac{1}{1 + \sigma T} \qquad \nu = \frac{b_1 T}{\Delta_x^2}
\end{aligned}$$

Difference scheme (3) is formally second order accurate in space and first-order accurate in time, due to the use of the operator δ_{t-}, necessary in order to obtain an explicit algorithm. Since the term involving δ_{t-} is in general a small perturbation, we do not expect such a term to have a deleterious effect on accuracy as a whole.

3.2.1. Stability

In the case of a plate of infinite spatial extent, a condition sufficient for stability can be determined with relative ease through the use of spectral or von Neumann techniques. It is given by

$$\frac{\Delta_x^2}{1 + \alpha^2} \ge \frac{2b_1 T + c^2 T^2 + \sqrt{(2b_1 T + c^2 T^2)^2 + 16\kappa^2 T^2}}{2} \quad (5)$$

It is important to point out, however, that when boundary conditions are applied, this stability condition becomes merely necessary; further analysis is required to determine whether a given boundary condition applied to a given finite difference scheme degrades the stability bound given above. We make no further comment on this other than that such analysis can be extremely delicate in general, and that we have observed no problems in the application of at least simple boundary conditions to this scheme.

We have phrased the stability condition as a bound on Δ_x and Δ_y (through α), the grid spacings, since for audio applications, it is natural to have the sample rate ($1/T$) be fixed from the outset. The purpose of having Δ_x distinct from Δ_y is so that for rectangular plates, it will possible to divide the lengths L_x and L_y into an integer number of parts, i.e., we will have $\Delta_x = L_x/N_x$ and $\Delta_y = L_y/N_y$ for integer N_x and N_y. This being said, it is advantageous (from the point of view of reducing numerical dispersion) to set α as close to 1 as possible (i.e., it is not advisable to set $\alpha = L_x/L_y$) and Δ_x as close to the bound given by (5) as possible.

3.2.2. Initialization

Difference scheme (4), second order in time, requires the initialization of the grid function $u_{i,j}^n$ at time steps $n = 0$ and $n = 1$, in terms of initial conditions $u(x, y, 0)$ and $\frac{\partial u}{\partial t}(x, y, 0)$. We can immediately set

$$u_{i,j}^0 = u(i\Delta_x, j\Delta_y, 0)$$

but the setting of $u_{i,j}^1$ requires slightly more care. Though there are many ways of performing this ini-

tialization, a simple way of proceeding is to set

$$u^1_{i,j} = u(i\Delta_x, j\Delta_y, 0) + T\frac{\partial u}{\partial t}(i\Delta_x, j\Delta_y, 0)$$

which is a first order accurate approximation. Higher order accurate initialization may be accomplished by introducing spatial derivative information.

It is important that the setting of the initial values of the grid function not conflict with values prescribed by the forcing function f; in other words, if both initial conditions and an excitation are present, it is desirable that the excitation not come into play until after the first two time steps.

3.2.3. Boundary Conditions

As mentioned above, we can arrange that $L_x/\Delta_x = N_x$ and $L_y/\Delta_y = N_y$, meaning that our grid function $u^n_{i,j}$ is delimited to indices $i = 0, \ldots, N_x$, $j = 0, \ldots, N_y$. As updating difference scheme (4) requires, at a given grid point, access to values of the grid function at the previous time step at most two spatial steps away in the x or y directions, we need to modify the scheme at points on the boundary, or one step away, i.e., for $i = 0, 1, N_x - 1, N_x$, for all j and $j = 0, 1, N_y - 1, N_y$ for all i.

Considering the fixed conditions (2), it is clear that the condition $u = 0$ may be enforced simply in the difference scheme by setting $u^n_{i,j} = 0$ at grid points directly on the boundary; it is thus not necessary to compute these values, or even store them during the simulation. Now consider a grid point adjacent to the boundary, such as i, $j = 1$. Clearly, from (4), updating $u^n_{i,1} + 1$ will require access to $u^n_{i,-1}$, which is not included on the grid. Finite difference approximations to the derivative boundary conditions (2) yield, however,

$$u^n_{i,1} = u^n_{i,-1} \quad (6)$$

$$u^n_{i,1} = -u^n_{i,-1} \quad (7)$$

In order to apply these boundary conditions, one may begin from (4), and simply replace any instance of a grid variable from beyond the boundary by one of the conditions given above.

We do reiterate, however, that the implications for stability are by no means trivial, i.e., a difference scheme which is stable over the interior of the problem domain can become unstable under the application of boundary conditions; further analysis, which may be extremely complex, is required.

3.2.4. Input/output

One means of exciting the plate is through its initial conditions; another is through the use of a driving function $f(x, y, t)$. For interesting audio results, it is useful to consider a moving point source, i.e., $f(x, y, t) = g(t)\delta(x - x_f(t), y - y_f(t))$; here, $x_f(t)$ and $y_f(t)$ are the parameterized coordinates of the source of strength $g(t)$ at time t. Suppose, at some given instant $t = nT$, we have $i^n_f = \lfloor x_f(nT)/\Delta_x \rfloor$ and $j^n_f = \lfloor y_f(nT)/\Delta_y \rfloor$ and $\epsilon^n_{x,f} = x_f(nT)/\Delta_x - i^n_f$ and $\epsilon^n_{y,f} = y_f(nT)/\Delta_y - j^n_f$, the source may be bilinearly spread to four neighboring grid points by

$$\begin{aligned}
\hat{f}^n_{i^n_f, j^n_f} &= (1 - \epsilon^n_{x,f})(1 - \epsilon^n_{y,f})g(nT) \\
\hat{f}^n_{i^n_f+1, j^n_f} &= \epsilon^n_{x,f}(1 - \epsilon^n_{y,f})g(nT) \\
\hat{f}^n_{i^n_f, j^n_f+1} &= (1 - \epsilon^n_{x,f})\epsilon^n_{y,f}g(nT) \\
\hat{f}^n_{i^n_f+1, j^n_f+1} &= \epsilon^n_{x,f}\epsilon^n_{y,f}g(nT)
\end{aligned}$$

One must take care that the source coordinates do not fall within a spatial increment of the boundary, otherwise a more complex (asymmetric) spreading strategy will become necessary. Other higher-order source spreading strategies are of course possible.

A scalar output signal u^n_o at moving coordinates $x_o(t)$, $y_o(t)$ can be treated similarly, through bilinear interpolation of the computed output values $u^n_{i,j}$. I.e., if we set $i^n_o = \lfloor x_o(nT)/\Delta_x \rfloor$ and $j^n_o = \lfloor y_o(nT)/\Delta_y \rfloor$ and $\epsilon^n_{x,o} = x_o(nT)/\Delta_x - i^n_o$ and $\epsilon^n_{y,o} = y_o(nT)/\Delta_y - j^n_o$, then one can obtain

$$\begin{aligned}
u^n_o &= (1 - \epsilon^n_{x,o})(1 - \epsilon^n_{y,o})u^n_{i^n_o, j^n_o} + \epsilon^n_{x,o}(1 - \epsilon^n_{y,o})u^n_{i^n_o+1, j^n_o} \\
&+ (1 - \epsilon^n_{x,o})\epsilon^n_{y,o}u^n_{i^n_o, j^n_o+1} + \epsilon^n_{x,o}\epsilon^n_{y,o}u^n_{i^n_o+1, j^n_o+1}
\end{aligned}$$

Obviously, as many different output signals as desired may be computed in this way.

A particularly nice spatializing effect is achieved by a rather slow rate of motion of the output locations relative to one another, which leads to interesting phasing effects; in a sense, this can though of as a very mild form of scanned synthesis [5] where the scan rate is slow in comparison with the rate of vibration of the system.

3.2.5. Sound Examples

An example of a sound output is given in Figure 1 below, under parameter choices as given in the caption. In this case, the grid size is 26× 34, and the total time required to output 1 s of sound, at 44.1 kHz, is approximately 22.6 s on a 2 GHz workstation. This corresponds to a relatively low-pitched sound (lowest frequency 72 Hz). For smaller plates, or plates of different thickness (reflected in the value of κ), the grid sizes and thus the computation time will be significantly faster.

3.2.6. Computational Complexity

In the general case, for $\Delta_x \neq \Delta_y$, scheme (4) requires nine multiplies per grid point, per time step. For each time step, updating the entire grid will require roughly $9\alpha L_x L_y/\Delta_x^2$ multiplies, and thus the number of multiplies per second required will be $9\alpha L_x L_y/\Delta_x^2 T$.

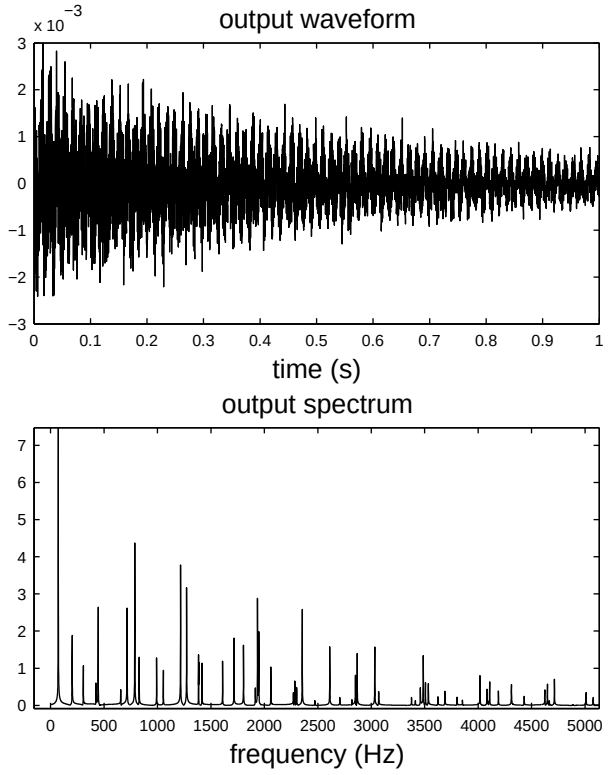

Figure 1. *Sound output (top) and spectrum (bottom) for a clamped steel plate ($\kappa = 15.26$), with side lengths $L_x = 1m$ and $L_y = 1.3m$, excited by an initial condition in the form of a raised 2D cosine, centered at the plate center, with a width of 20 cm. The output is read from a point at coordinates $x = 5L_x/8$, $y = L_y/2$. Sample rate is 44.1 kHz.*

Considering the simplest case of $c = b_1 = 0$, and assuming that (5) is satisfied with equality (which is nearly true in practice), we will then have $9\alpha L_x L_y/2(1+\alpha^2)\kappa T^2$ multiplies per second. For a square steel plate of side length 1 m and of thickness 2 cm, and assuming that $\alpha = 1$, this amounts to approximately 1.4×10^8 multiply operations per second, which, while by no means cheap, is within the capability of a gigahertz-range processor.

Direct finite difference modelling of musical instruments is often characterized as being more computationally demanding than other techniques, in particular modal synthesis. It is important to note the following: a finite difference scheme for a linear time-invariant system can always be written in state-space form, the number of modal frequencies being half the size of the state; in this sense, the finite difference scheme behaves as a modal synthesizer. Traditional modal synthesis requires approximately 2 multiplies per mode, per time step (if a mode is to be implemented as a two-pole resonator). The finite difference scheme may require more, but not significantly so...in the present case, 6 multiplies per grid point, per time step, where the number of grid points scales directly with the number of modes computed. There are advantages and disadvantages to either approach: modal synthesis deals with simple geometries and boundary conditions well, and is applicable only to LTI systems. Finite difference schemes suffer none of these restrictions, but additional considerations, such as numerical stability, and dispersion, can become problematic.

4. CONCLUSIONS AND FUTURE DIRECTIONS

Though the algorithm presented here is not computationally cheap, the calculation costs are of the same order of magnitude as modal synthesis [1], and is more general, in the sense that a) it does not rely on precomputation of of modal frequencies and shapes (which can be very expensive in terms of memory storage requirements), and b) it may be extended to model the more musically interesting nonlinear plates (perhaps of the von Karman variety [6]); nonlinear systems are not well-modelled by modal techniques. Another important issue is that of spatialization when a three-dimensional array of speakers is available (as is the case in the Sonic Lab, at SARC, in Belfast).

5. REFERENCES

[1] J.-M. Adrien. The missing link: Modal synthesis. In G. DePoli, A. Picialli, and C. Roads, editors, *Representations of Musical Signals*, pages 269–297. MIT Press, Cambridge, MA, 1991.

[2] S. Schedin, C. Lambourg, and A. Chaigne. Transient sound fields from impacted plates: Comparison between numerical simulations and experiments. *Journal of Sound and Vibration*, 221(32):471–490, 1999.

[3] C. Lambourg, A. Chaigne, and D. Matignon. Time domain simulation of impacted plates. part ii. numerical model and results. *Journal of the Acoustical Society of America*, 109(4):1433–1447, 2001.

[4] K. Graff. *Wave Motion in Elastic Solids*. Dover, New York, New York, USA, 1975.

[5] B. Verplank, M. Mathews, and R. Shaw. Scanned synthesis. In *Proceedings of the 2000 International Computer Music Conference*, pages 368–371, Berlin, Germany, 2000.

[6] C. Touze, O. Thomas, and A. Chaigne. Asymmetric nonlinear forced vibrations of free-edge circular plates. part i. theory. *Journal of Sound and Vibration*, 258(4):649–676, 2002.